

AU/ACSC/184/1999-04

AIR COMMAND AND STAFF COLLEGE

AIR UNIVERSITY

OPEN SOURCE AND THESE UNITED STATES

by

C. Justin Seiferth, Maj USAF

A Research Report Submitted to the Faculty

In Partial Fulfillment of the Graduation Requirements

Advisor: Lt Col Anida Wishnietsky

Maxwell Air Force Base, Alabama

April 1999

Disclaimer

The views expressed in this academic research paper are those of the author(s) and do not reflect the official policy or position of the US government or the Department of Defense. In accordance with Air Force Instruction 51-303, it is not copyrighted, but is the property of the United States government.

Contents

	<i>Page</i>
DISCLAIMER	ii
PREFACE	v
ABSTRACT	vi
INTRODUCTION.....	1
JUST WHAT IS OPEN LICENSING?.....	5
Definitions.....	6
The Culture	8
The Institutions	10
WHY US? THE INFORMATION ECOSPHERE & DOD	14
APPLYING OPEN LICENSING TO GOVERNMENT SOFTWARE ACQUISITION AND MAINTENANCE.....	27
Commercial Off the Shelf (COTS)	27
Customized Packages.....	31
Customized Automated Information Systems	32
Military Unique Development	34
SUCCESS STORIES	37
How Open Licensing Breeds Success.....	39
THE LAST HALF MILE.....	42
THE OPEN SOURCE DEFINITION (<i>VERSION 1.0</i>)	48
1. Free Redistribution.....	48
2. Source Code	48
3. Derived Works	49
4. Integrity of The Author's Source Code	49

5. No Discrimination Against Persons or Groups.....	49
6. No Discrimination Against Fields of Endeavor.....	49
7. Distribution of License.	49
8. License Must Not Be Specific to a Product.	50
9. License Must Not Contaminate Other Software.....	50
10. Example Licenses.	50
ADOPTION OF OPEN LICENSING.....	51
Phase 0: Concept Exploration.....	51
Phase 1: Program Definition and Risk Reduction	53
Phase 2: Engineering and Manufacturing Development	54
Phase 3: Production, Fielding/Deployment and Operational Support.....	58
Operational Support.....	60
GLOSSARY.....	64
BIBLIOGRAPHY	66

Preface

I have long been an active participant in the development of software for use by the government and others on the internet. Given this interest, I welcomed the opportunity afforded me during my stay at Air Command and Staff College to examine how the ideas of open licensing may be applied to our work and of benefit to both the Air Force mission, the Department of Defense and hopefully the taxpayers as a whole.

As a result of the success of the Open Source Movement and the World Wide Web spawned by its products, you'll notice that the bibliography of this paper relies heavily on periodicals and titles which you won't find in the Air University catalogue. I thank the many people who have taken the time to post their information and opinions on the internet for the enrichment of others. I owe a debt of gratitude to the Air University librarians for helping with the many interlibrary loan requests I made.

There's a Chinese saying that the philosophy of one century is the common sense of the next. There are many controversial ideas in this paper; ideas which may at first seem foreign and even counterintuitive. Much of the work in this area is based on the personal beliefs of a few outstanding individuals. I'll let the validity of their beliefs stand on the success of their results and revolution they have sparked.

Abstract

Over the past 40 years a collective form of systems development has evolved on the electronic networks of the world. In the wake of the information technology revolution has come a proven method for developing, deploying and maintaining these systems. This method, developed under the auspices of Department of Defense research grants, has resulted in the most successful and reliable software in existence.

This method, based on collective intelligence, peer review and functional evolution, has rippled through the world of Information Technology. It depends on the uninhibited distribution of the currency of this realm: the source code, documentation and data which are the building blocks of these complex systems. The release of source code is commonly called open source licensing. The release of electronic information is known as open content licensing. Together, they comprise Open Licensing.

There are significant gains to be realized through the formal adoption, support and use of open licensed systems by the Department of Defense. Secondary gains may be made in the morale and retention of Airmen involved in information technology. This adoption can take place at any point in the acquisition cycle and can even benefit deployed and operational systems. The benefits include reduced acquisition, development, maintenance and support costs and increased interoperability among our own systems and those of our Allies.

Chapter 1

Introduction

The capacity of the female mind for studies is of the highest order cannot be doubted, having been sufficiently illustrated by its works of genius, of erudition, and of science.

—James Madison
The Writings of James Madison

Seldom do those who innovate fully recognize what they have created. Moreover, it may be many decades before a new idea gains social recognition. Change, regardless of supporting evidence, is not easy for any society or industry to accept. For example, some twenty years before a young socialite named Ada Lovelace showed the world the power of computing, James Madison had recognized the capability of women.¹ Despite the recognition of female talent and intelligence by at least one founding father it was another century till women in the United States were granted suffrage. They still have not obtained parity in salary or representation within the fields of science and technology. New methods must first prove themselves among visionaries before they will be adopted by industry. The methods must then be highly successful within leading industries before they gain general adoption. Time is needed to build the required infrastructure and expertise; society must maintain the equities of older methods².

Widespread adoption of open licensing can greatly improve information systems are developed and distributed. This paper will discuss this set of ideas, methods and systems which may be new to some within the Department of Defense. The ideas, methods and systems resulting from open licensing will be applied to various aspects of software-intensive systems. The validity of this application will be examined and discussed. It's hoped that this analysis will reveal whether opportunities to utilize open licensing exist.

Open licensing is a term which embraces both software and electronic documentation. Over the last forty years open source software has proliferated and come to dominate many important market segments. These segments include the world's most popular world wide web and email servers, the fastest-growing operating system and the underpinnings of the Internet itself. Since being coined by Ms. Christine Peterson of the Foresight Institute, the term Open Source has become a hot topic in the commercial world; it's made headlines in major newspapers, the cover of Forbes and many other national magazines.³ Open Source is a big thing. Some believe that the government is on the tail end of emerging technical trends and new ideas.⁴ In this case, the Department of Defense made the initial investment in the technology but has yet to fully capitalize upon it.

Open licensing of source code and documentation is a legal construct like a copyright. Open licensing has been successful for precisely the same reasons as copyrights on written work- because they both promote the common good. Consider, prior to the invention of the printing press there was no need for copyrights. It was almost as much effort to duplicate a work as it was to write it. The printing press changed this by allowing the ready reproduction and distribution of

knowledge. Copyrights were enacted to allow the wide distribution of knowledge during the industrial age. Networking technology offers an opportunity to even more easily distribute the knowledge of the information age.

The industrial age required society to reorganize around the manufacture of sophisticated and abundant products. Software systems require as much labor to produce as any industrial age product. Unlike the warehouses needed for industrial age products, an electronic system, which cost millions of dollars to development, can be stored on a single compact disk. Unlike the railroads needed to distribute products, a software system can be sent through email or published on the web in the blink of an eye. The distribution of electronic information can be effortless and essentially free. This ease of distribution serves the public good by capitalizing on a strength of technology and increases efficiency by eliminating the storage and distribution costs which so burdened the products of the industrial age.

Software may be just information, but restrictions make the distribution of software neither effortless nor free. Many people regard software as a product, in the same category as a nail or a pail. Strangely though, unlike these items, the licensing terms of many software applications indicate that we cannot resell the software or even legally loan it to a friend. We can freely dull a nail to prevent it splitting a board. Yet, many commercial licenses prohibit software users from adding notes to its document or features to its implementation even if those changes would improve the value of the system to the customer. A pail can be used to carry water or as a planter, in either case we would not expect it to leak. If the software is used in a new situation, often the corporation holding ownership neither accepts responsibility nor offers to help get the

software working in the new environment. The current Y2K problem is only the most recent example of the expensive consequences of these license agreements.

Most people will agree that the cost of software stems from the need for its authors, and the businesses which are built around them, to make a living. This paper will look at an alternative way for all these people to make a living and for the government to get more for its money. The alternative is known as open licensing and it can have a big effect on the way the world does business. The derivatives of Open Licensing may also help alleviate some of the Government's difficulties, including personnel retention and recruitment.

Chapter 2

Just What is Open Licensing?

Back in the early days of punch cards & magnetic core memory virtually all software was open source. There was a lot to be written and not many people skilled at producing it. Progress in those days depended on making sure the best examples of code were available so developers could build upon what little already existed. Users of the software and other developers helped the original authors improve their code. These participants became a pool of knowledge available and interested in supporting and continuing the development of the system. If a developer used, enhanced or was inspired by someone else's work, they gave them credit by reference to their contribution. In literary terms, this system is very like the notion of footnotes. The system worked very well and is responsible to some extent for the early explosion of automation.⁵

Some situations haven't changed much since the 1960s. There's still a lot to be done and not very many good people available to do it. New developers still learn from those more experienced by taking up the maintenance of projects which others began. Within the open source community, this process of developing systems in a cooperative environment and sharing the results of a team's work with others, has been coined "the bazaar". The bazaar is the key process which has allowed users and developers, spread across the internet, to tackle huge,

sophisticated problems like operating systems. The same process allows these projects to advance far more rapidly and with greater quality than many traditional methods.

Definitions

A complete definition of the Open Source license framework is included as Appendix A. Rather than becoming mired in a technical definitions open licensing, it may be more instructive to focus on the effects of its use. Open licensing asserts that the redistribution of the documentation and software should not be unnecessarily restricted. Further, and this is the heart of the topic, that widespread peer review is the most effective method of finding and correcting faults. Two of the more influential figures behind open licensing, Mr. Raymond a spokesperson for the open source movement and Mr. Stallman, the founder of the Free Software Foundation, have their own ways of expressing these ideas. As Mr. Eric Raymond suggests, "Open Source promotes software reliability and quality by supporting independent peer review and rapid evolution of source code"⁶. Richard Stallman, the author of the most popular open source license, has a succinct way of describing open licensing: "Think free as in speech, not free as in beer". Virtually every American sees free speech as an inalienable right, but only the most optimistic among us expect Anheuser - Busch to distribute free Budweiser.⁷

These are the main tenets of the Open Source and Open Content licensing frameworks used to determine how the source code of software applications and content of published documents must be treated. Open source licensing generally refers to the conditions under which software

source code meant for use by computers is "released" or made available to those other than the author. Open content is generally understood to refer to text meant for human consumption.

For simplicity's sake, these two areas will be collectively described as "open licensed". Under open licensing, authors retain full rights to their works, but there are much simpler restrictions on the ways others can utilize the work. Like other important declarations, the consequences of these simple statements are broad and deep- all pervasive in fact.

Open licensing allows an author or corporation to retain ownership of a program or document while giving others the freedom to improve its quality or capabilities provided that they in turn agree to share their developments with others under the same terms. Content released under these terms allows a new class of user - developers to contribute to the success and quality of the products their livelihood depend on. Experts can refine, enhance and distribute new services to the extent of their own ability. Geniuses can feel free to innovate, create and inspire. This triad of user-developers, experts and genius forms a development community of hundreds of thousands or even millions of people. The large number of people involved allows for dynamism that's impossible under any other conditions. This change has sparked a revolution in the way and type of software that's developed and the industries which profit from it. The success of the internet is the success of open source development. The success of the world-wide web shows the power of open content. Open licensing affects the way software is developed, distributed, used and supported. It may change the way information technology focused corporations generate profits.

The Culture

In many ways, open source licensing is an extension of the peer review process at the heart of the scientific method. Pragmatism demanded open source during computing's early days. Ideals and beliefs, not formulae or science were the seeds of the open source resurgence. Like scientists, open license proponents insist that software and documents stand up to the strictest scrutiny. This scrutiny is not a simple process of running an application against test cases or having a few colleagues skim a text; careful analysis of source code and documents is required. The review must examine design as well as functionality, the structure as well as the content. As Admiral Grace Hopper observed, ".. programming is more than an important practical art. It is also a gigantic undertaking in the foundations of knowledge." ⁸

The ideas behind open licensing are fundamental to the origination of science and extend back to the 17th century renaissance.⁹ Scientists have always published their results for others to examine, verify and use. Just as many early scientists were also artists, architects and business people in the early days, early software developers and users were often one and the same. When software pioneers came up with a particularly efficient implementation of an algorithm or a new idea, they often gave a copy of it to their friends and associates for them to use and expected the same courtesy in return. Within this system, productive developer-users gained a certain element of notoriety. They found it easier to get jobs and enjoyed a celebrity status among their peers. Their employers became rich and famous also. International Business Machines, the Massachusetts Institute of Technology and later Sun Microsystems, Cisco Systems and recently Yahoo! Corporation have all contributed to and benefited from this system.

Everyone who contributes in this process learns how to work smarter as their work is scrutinized by some of the best minds in the business, the collective result of everyone's efforts produces something greater than any one individual could produce or afford. New ideas, such as the Internet emerged and were widely implemented using the easily obtained source code. New ideas and implementations could be examined and discussed. Suggested changes to these ideas could be sent back to the original author of the code¹⁰. While funds may not have been exchanged between the originator of the software and the eventual users everyone benefited from the arrangement. The essential ingredient for this exchange has always been the ability to communicate. The quality and productivity of systems developed in this way depends upon ready and reliable communications.

Just as the internet has increased the ability of people to exchange information and ideas raising the floor of economic activity and ideas: it has also increased the capability of open source development teams. As the internet expands, so does the number and productivity of the open source development teams. In some ways, these open license development techniques are the ultimate extension of Microsoft's attempts to scale loosely structured teams.¹¹ Open source teams can be expanded indefinitely allowing progress to take place around the world, 24 hours a day.

Products often reflect the personality of the organization which developed them and open license projects are no different. For instance, collaborative nature of development is captured in the systems themselves. Software developed under open source licensing combines what have

often been thought to be conflicting goals; innovation, reliability and interoperability. The internet community values efficiency greatly- that is they do not like to redo what has been accomplished. As such most open licensed software incorporates other, existing open licensed code and utilizes only a few common data formats. As a result of constant reuse and refinement, the open source community has reduced the need for extensive support of its offspring. The support which is offered often comes from the original developers and extends not only for an extended period but as you would expect is of very high quality. The industry and press has recognized these benefits. In 1998, the readers of *Infoworld*, a major information technology weekly, awarded open source operating system both the "Best Operating System" and "Best Technical Support" awards for the second year in a row.¹² One open source word processor is so reliable the author will pay you \$327.68 to report a bug.¹³ The storage format, templates and other data used by this particular word processor have remained constant for over a decade, yet the application has maintained or exceeded the features of its commercial competition.

The Institutions

Open source has received comparatively little official attention from the organization that funded most of its early production. As the worlds' largest information technology customer this organization stands the to gain much from it's proliferation. The organization is, of course, the Department of Defense.

Within the Department of Defense, the National Laboratories and Defense Advanced Research Agency have been the most visible users and producers of open licensed systems.

They've released such advances as the original firewall and network security toolkits. As a more recent example, within the last year the National Air and Space Agency has debuted several inexpensive supercomputers. Open licensed operating systems and applications allowed the scaling of inexpensive pentium-based machines into an integrated hardware/software system. In addition to being inexpensive, these machines are among the most powerful available.¹⁴

The Department of Defense may have initiated much of the research which spawned open source development but commercial organizations have been quick to recognize the opportunities the methodology offers. During the past few years, an increasing number of commercial enterprises have begun offering open source systems as products. These corporations include Netscape Communications, Cygnus, Sendmail, McAfree, Caldera and Red Hat Software. Traditional vendors such as Hewlett-Packard, Oracle, Dell, Compaq, Silicon Graphics, Sun Microsystems and International Business Machines (IBM) have also begun to resell and support open source systems.¹⁵ Open licensed systems are generally introduced to an organization by the engineering divisions. In fact, the entire movement is a product of technologically oriented individuals primarily motivated by other than economic goals. Once implemented however, the business sectors have recognized the productivity and cost savings aspects of the technology. Business is particularly interested in the opportunity to reduce administrative and production overhead allowed by the development process typically used for open licensed systems.

Open source development uses ad hoc organization and electronic communities. This set of communities is commonly referred to as the "bazaar".¹⁶ The term "bazaar" is used in reference

to the bazaars of past years where vendors, specializing in certain commodities, would gather together in open air markets to fulfill their community's cumulative needs. Many organizations such as the Apache Group, developed the most popular world wide web server without ever having met in person. Other organizations such as Berkeley Software Design Incorporated, maker of the trusted operated system upon which many commercial firewalls are based, do not exist, in the traditional sense. While they have a headquarters in Colorado, the corporation has employees distributed in many areas of the US who collaborate electronically.¹⁷

The bazaar is a basic institution of open licensing. The term refers to an organization style based on survival of the fittest and most efficient. The bazaar binds the developers, marketers, writers and users of open source together in an interactive, electronic web. The bazaar works to ensure that the "best" products, as judged by a majority of interested parties, survive. The availability of source ensures that nobody is disenfranchised. If an application or data format is important to an organization or individual they are free to continue using, maintaining and enhancing the product. They are also free to incorporate and integrate any aspect of the technology into future offerings so that the return on their development, training and maintenance investment is maximized through long use and reuse. This process results in systems which harnesses the collaborative abilities of its user developers to create products of equal or greater quality than any produced by traditional means.¹⁸ This area will be discussed and applied to various type of software products in the next chapter.

¹ Dr. Betty Toole, "Ada Byron, Lady Lovelace", *The Ada Lovelace Web Site*, n.p., on-line, Internet 10 November 1998, available from <http://www.scottlan.edu/lriddle/women/love.htm>

-
- ² Thomas P. Hughes, *American Genesis*. (Middlesex, England: The Penguin Group, Harmondsworth, 1989), 460
- ³ Fortune Magazine, 10 August 1998, Cover
- ⁴ Clemens, Admiral. "IT-21: The Path to Information Superiority, Quadrennial Defense Review Secretary's Report to Congress", *Chips*, July 1997, n.p., on-line, Internet, 11 November 1998, available from http://www.chips.navy.mil/chips/archives/97_jul/file1.htm
- ⁵ Mark Leibovich, "The Spreading Grass-Roots Threat to Microsoft", *Washington Post*, December 3, 1998, A01
- ⁶ Eric S. Raymond, "Frequently Asked Questions about Open Source", *The Open Source Web Site*, n.p., on-line, Internet 11 November 1998, available from <http://www.opensource.org/>
- ⁷ Tim O'Reilly and Ether Dyson, "Open Mind, Open Source, Release 1.0", *O'Reilly Publishing Corporation Web Site*, n.p., on-line, Internet 2 January 1999, available from <http://www.oreilly.com/>
- ⁸ Ibid.
- ⁹ James Burke, *The Day the Universe Changed*. (Boston, MA: Little Brown & Company, 1986), 153
- ¹⁰ Donald E. Knuth, "George Forsythe and the Development of Computer Science". *Selected Papers on Computer Science*. (Stanford, CA: Center for Study of Language and Information Publications, 1996) 241
- ¹¹ Michael Cusuman and Richard Selby, "How Microsoft Builds Software", *Communications of the ACM*, Jun 1997, 53-61
- ¹² Edward Foster, "1997 Product of the Year", *Infoworld*, n.p., on-line, Internet, 20 December 1998, available from <http://www.infoworld.com>
- ¹³ Donald E. Knuth, "Rewards", n.p., on-line, Internet, 20 December 1998, available from <http://www-cs-staff.stanford.edu/~knuth>
- ¹⁴ Jonathon Ewing and Sami Lais, "70 CPUs add up to big power", *Government Computer News*, 7 September 1998, 43
- ¹⁵ David Pendery, "News", *Infoworld*, 1 Feb 1999, 16
- ¹⁶ Eric S. Raymond, "The Cathedral and the Bazaar", *The Open Source Web Site*, n.p., on-line, Internet 11 November 1998, available from <http://www.tuxedo.org/~esr>
- ¹⁷ "Company Information", *Berkeley Software Design Incorporated Web Site*, on-line, Internet 20 March 1999, available from <http://www.bdsi.com>
- ¹⁸ Vinod Valloppillil, "Open Source Software, A (New?) Development Methodology", *The Open Source Web Site*, on-line, Internet 11 November 1998, available from <http://www.tuxedo.org/~esr/halloween.html>

Chapter 3

Why Us? The Information Ecosphere & DoD

A popular Government without popular information or the means of acquiring it, is but a Prologue to a Farce or a Tragedy or perhaps both. Knowledge will forever govern ignorance, and a people who mean to be their own Governors, must arm themselves with the power knowledge gives.

—James Madison

When people shop for shoes they don't just pick a pair that fits them, the shoes have to look good also. The government, of course, is usually more concerned with function than style and information technology isn't a pair of shoes, but it illustrates an important point. Just because a good idea seems to offer benefits, or fits well with an organization's beliefs doesn't necessarily mean a course should be pursued. There are many organizations and processes who might be impacted by a switch to open licensing, so we should have some compelling reasons to expand its use. This paper will first see if open licensing can fit into the government's way of doing business. If the fit seems good, then it becomes important to determine what benefits and costs open licensing might bring to the government. If the fit seems good and the benefits outweigh the costs then, as the saying goes, if the shoe fits, wear it.

The government is well suited to the adoption of open licensing because of its historical traditions and legislation, the many non-monetary exchanges which characterize interagency cooperation and the government's disparate and distributed organizations. Unlike many other

nations, the United States has been characterized by a relatively transparent government; meaning a society which freely shares information with its citizens and believes the exchange of information and goods contributes to the public good.¹ Our founding fathers and more recent legislation demand the widespread dissemination of public information and capabilities among citizens and allies. It is a fundamental principle of our culture to encourage the free exchange of information for government and commercial purposes.² Open licensing of our software and documentation is a natural extension of this characteristic.

The cooperative and non-monetary remuneration used within open licensed systems development may also fit well within a government setting. Like companies, the various parts of the Department of Defense must cooperate to produce joint systems. Unlike commercial organizations, agencies within the Department of Defense rely on barter and management agreements rather than monetary exchanges. The government's system makes enforcement of agreements difficult as there is often no binding contract or clear higher authority to appeal to for a decision. The use of open licensing can have a role in encouraging cooperation on information intensive systems. The availability of source prevents information hoarding and encourages cooperation. Under open licensing conditions, only through contributions can developers, programs and organizations expect to receive help in turn from others. This doesn't obviate the power of money or traditional motivational methods but offers an additional dimension which can be used to motivate and retain personnel.

As Sun Tzu observed, it is difficult to place a price tag on the value of leadership or need of followers to be appreciated by their leadership and peers; loyalty and dedication are seldom

purchased.³ The traditional military concepts of "service before self" and "excellence in all we do" are appreciated and practiced widely within the open source community. Sun Tzu, the ancient Chinese war theorist whose writings are well known within the military, is widely read and quoted throughout the open source community. In fact, his sentiments are at the heart of motivation and practice of its proponents. Open source practitioners work long and hard on their labors of love or need, it requires true leadership skills to organize them to do so.

The creation of bazaars to develop open licensed systems may give government and commercial IT specialists an incentive beyond dedication to stay within the Department of Defense. Budgetary restrictions prevent the government from offering the fiscal benefits of many private industries. The starting salaries of Computer Scientists exceed \$60, 000 in many areas of the country.⁴ While defense contractors have greater flexibility than government agencies, many of them are also no longer able to offer the compensation found at commercial corporations. In addition, it is not clear that the military is still regarded as a source of cutting-edge interesting work. Fortunately it is clear that remuneration is just one of many aspects affecting developer retention. Many developers and administrators place greater value on the opportunity to do interesting work.⁵ Open source development projects are regarded as challenging work offering the extra benefit of being able to obtain recognition by sharing the full extent of your talents with others. Many open source projects are able to use this effect to achieve excellent retention rates. The Apache web server team for instance, has maintained five of the original eight core developers for over last four years.⁶ Cygnus, the largest privately held corporation based on open

licensed projects, experiences turnover rates less than a tenth that of other Silicon Valley firms. Cygnus founders credit much of their success to the attractive power of open licensed projects.⁷

The adoption of open source licensing may allow the military to leverage some of the current enthusiasm garnered by open license related methods. Open licensing offers an alternative for developers who are frustrated, feeling that they have wasted their time on projects which were not released for their intended purpose, but still contained valuable segments which might have been reused if licensing permitted.

Allowing developers to share their work gives them a physical portfolio which adds to their value, productivity and sense of accomplishment. When developers do change jobs, the Government can expect them to bring their portfolio with them. This "reuse" lowers development costs in a real way. The portfolio serves as a discriminator by providing employers with tangible evidence of the talent or lack thereof of potential employees. The judicious application of open licensing offers the possibilities of improving both the performance of government systems and the job satisfaction, competence and retainability of military members, civilians and contractors.

Just like many organizations developing open licensed systems, the United States Government is deployed world wide. Multinational programs and interoperable systems developed by multiple contractors will impose new demands on the Government's acquisition system. As functions such as logistics change from personnel-intensive to information-intensive activities, greater levels of system and organizational interoperability will be required. Fewer government & contractor employees will be required to bring together more complex systems in

less time. The traditional approach of standards and specifications or purchasing all types of software from a single vendor may be able to produce workable systems, but experience has shown this approach to be a difficult and expensive path.

The lessons learned on the internet indicate that requirements selection and testing⁸ are more effectively accomplished by decentralized management, Darwinist selection, reutilized source code and simple protocols encouraged by open license development.⁹ Open licensing avoids nonessential, insecure, unwanted or unreliable functions. Traditional systems are designed to "get it right" out of the box. Open license applications start out weak in functionality and allow the marketplace to evolve them. For this reason, open licensing favors systems which produce results even when their eventual functionality is not fully implemented.¹⁰ As with people, these applications can evolve and move to new environments.

The government places great stock in commonality as a tool to reduce maintenance, administration and development costs. This concern is well placed as studies have shown lack of commonality to be a major cost driver.¹¹ Training is a major part of that cost. It is very difficult to train users, administrators and developers. Once trained, they become extremely valuable even mission essential assets. The government experiences very high turnover rates among military and contracted personnel. On any project, the cost of changing personnel is high and the incidence of personnel changes in today's economy higher still. Turnover rates range up to 70% with averages¹² of 35 - 45% per annum.¹³

It is possible that the development of open license systems may help alleviate project delays resulting from the Government's retention difficulties. The government has difficulty keeping information technology positions staffed and many projects, both contracted and internal, suffer greatly when a leadership change disrupts the momentum of ongoing projects. Open license projects are often perceived as lacking a "central authority". Ironically, they are heavily dependent on the existence of a recognized leader. Leadership ability is critical since there is no monetary incentive for many open source developers. Participation depends on the bond between the project leader and other participants. Unlike many traditional organizations, the leader is not a stagnant position, its occupant can vary according to talent and desire. The loose connections between leader and participants allows decentralized open licensed projects to handle leadership and participant changes well. As a result, they are more robust than centralized development teams or projects under the strict control of a single manager. The reasons for this strength are many but central is the ability of all participants to freely access the source code and keep abreast of progress. There can be no information hoarding on an open source project, they are not as vulnerable to familiar problem of a developer keeping others from reviewing their progress or project leader keeping bad news under wraps.

Having established that the concept of open licensing appears to fit well with some basic beliefs of our culture, it's important to move on to the second question posed at the beginning of this chapter. What does the government stand to gain by open licensing? Are there tangible benefits that we cannot otherwise obtain?

The three primary benefits the government can obtain are interoperability, improved long term access to data and an improved ability to incorporate new technology. Interoperability favors the simple, standardized protocols and data formats generally used by open source applications. Much of the effectiveness of the United States military has been gained through its pioneering efforts to promote interoperability and exploit the advantages interoperability allows. Specifications may promote interoperability but experience has shown they do not ensure it.¹⁴ Extensive testing increases the likelihood that systems will interoperate. However, as more systems and their interpretations of specifications are integrated, the difficulty of ensuring interoperability among all combinations grows exponentially. The number of variables in this scenario may be reduced by incorporating the same code within all systems. That is, to use the same interpretation of a specification in all applications. Open licensing allows you to do precisely this. Once a workable implementation of a specification exists, open licensing permits an organization to utilize that very same code, documentation and data formats into every component of the overall system.

Consider the government's need to maintain long-term access to archived information. In addition to the changes in physical media the protocols and formats used by applications can change dramatically over the life of a system. Data formats of non-open licensed systems seldom remain the same. For instance, the data formats of Microsoft's Word 6.0/95, 97 and 98 are all incompatible and indistinguishable from each other to humans, you must attempt to open a document to discover that it is corrupted or unreadable. By contrast, during the same period the Hypertext Markup Language used on the World Wide Web has evolved from version 2.0 to

version 4.0 yet all versions are interoperable and can be read with even the simplest text editor.¹⁵ The Hypertext Markup Language format is also many times more bandwidth efficient than an equivalent Microsoft Word format document. The point here is not to argue which format is superior, but to note the emphasis open licensed projects and specifications place on true interoperability, error recovery and efficiency. There is temptation to just use the latest office application storage format and convert the archived data into a readable format when it needs to be retrieved. How can archived information be retrieved when contemporary programs are no longer able to import it?

Consider how open licensing might affect data access. Today, an organization will often have to abandon years of work to use a new system or application. Under open licensing, an organization has full access to its own systems. Under open licensing, potential support vendors have access to the same information as the original supplier. This access means it's possible to contract out maintenance and enhancement to any vendor willing to become familiar with the systems. Differentiation under these circumstances is obtained by superior knowledge of the system. When the time to migrate from existing systems comes, the availability of open licensed information and the number of knowledgeable vendors will allow new systems to be adopted more easily. In the best case, a significant portion of the existing data, source code and documentation can be moved directly to the new environment.

Unfortunately, there are many cases such as this where there is no working code able to interpret the data and no information on how the data itself is formatted. The data may be in the unintelligible and easily corrupted binary formats favored by many modern office applications.

Once damaged or unreadable by contemporary applications, there is no partial recovery, this data is lost forever. These office applications represent the lower end of complexity; things become more difficult when you consider information stored in databases or other complex systems. If source code and standardized data formats and protocols are used however, gaining access to archived information becomes a matter of compiling or porting the access routines to contemporary hardware and converting the data, if necessary, to a contemporary format. Only when the source code and information on the format the application used is available, can we truly insure long-term access to the information.

As more and more information is available solely in electronic format, the information access situation may become more acute. In an effort to reduce expenditures, the government is moving to contract more business to private concerns. These vendors derive revenue from frequent releases. The update schedule can overwhelm the government's ability to integrate and distribute the updates. The asynchronous nature of commercial updates also disrupts the government's ability to plan for the change and incorporate new systems according to its own schedule. The era of automatic deployment of new releases is over, there must be some value added in the release to justify the cost of the upgrade. As a result, the government may react negatively towards vendor releases which don't bring any increased reliability or bring required features government customers demand. For instance, the United States Air Force was unable to keep many of its base level telephone switches up to date and as a result many switches were not Y2K compatible. When the impending millennium forced Y2K compliance the USAF needed to dedicate some \$70 million to fund an update and replacement of switches Service

wide. In the end, the contractor's revenue is diminished because they cannot sell the government as many updates as they are able to produce and the government is dissatisfied because they feel they are unable to keep up with the latest product release and must incur large costs for retraining and equipment upgrades when they do incur a major update.

Open licensing can lower the cost and risk of change- allowing the government to more easily adopt new technology. Open licensed project updates tend to be evolutionary and hence less disruptive to ongoing operations. In addition, since there is little economic incentive towards releases for the sake of revenue, updates to open licensed projects generally occur for a specific reason- either to fix errors, ease maintenance or incorporate features which have been specifically requested by users. Open licensing dramatically lowers the disincentives to evaluating new functionality, the scheme allows the government to try out a system without up front acquisition costs or contract actions. Vendors, in turn, gain an entrée to the governmental market and valuable feedback on the quality and suitability of their products. If generally satisfied with the system, the government can fund or the contractor provide small modifications or additions as required to fully meet requirements.

Modifications to open licensed systems may also come at a lower than traditional cost due to reduced "fog and friction". Without the usual restrictions on documentation or source distribution, many more developers will be familiar with the inner workings of the application so bugs are more easily unearthed and enhancements can be more rapidly incorporated.

It is reasonable to expect the government and contractors to adopt something as different as open licensing? The government has historically used applications and systems much longer

than commercial industry can afford to support them. With an annual IT budget exceeding \$57 billion, the government is still a large monolithic culture with a significant market presence.¹⁶ The wishes of the government are still a powerful influence on industry. Obviously, vendors will have to determine for themselves if such techniques would be profitable. However for those organizations "on the edge" of moving in this direction, the active or tacit approval of the government may provide the additional market incentive which convinces them to release their systems under an open license.

Within the open source community, there is an inherent tendency for a single implementation and a recognized owner of the official version to prevail. It is true that free access to information and a lack of discipline might allow multiple versions of documentation and systems to develop. Centralized development facilities and a preference for a single support vendor are some of the ways this problem is addressed, but avoiding the problem ultimately relies on organizational discipline. Discipline on the wild-west of the internet comes from a desire by developers to maintain their reputation. Claiming code developed by another to be your own work is a sure fire way to lose that reputation. It's far easier to give credit where credit is due. So rather than create divergent implementations, contributors tend to send their enhancements to the originator or "maintainer" of the application. The maintainer, in turn, incorporates the best of these changes into the core distribution. This arrangement benefits all. The community of users gets an enhanced application. The maintainer gains recognition and additional income through his or her association with the successful application and the organization or individual that contributed the change still has use of the enhanced application.

Some measure of recognition and income opportunity may result from individual contributions and those modification may in turn be enhanced or improved by others' effort. Within the military each component or particularly talented branch might contribute that which they know best. Contractors can contribute in the same manner to their own or other systems. This collaboration can function across both time and space bringing together many different interests.

Adopting open licensing seems to fit well within the government culture and bring with it some important benefits. How though can we actually begin to use it?

¹ "House Report 103-104 1993 Edition, House Committee on Government Operations", *The Library of Congress Web Site*, n.p., on-line, Internet 20 November 1998, available from <http://www.loc.gov/>

² "Leahy - Ashcroft Amendment to S.1609 to Authorize the Next Generation Internet", 26 June 1998, *The Library of Congress Web Site*, n.p., on-line, Internet 15 January 1999, available from <http://www.house.gov/>

³ Sun Tzu, *The Art of War*. (London UK: Oxford University Press, 1971), 35

⁴ Thomas York, "1999 IT Career Outlook", *Infoworld*, 30 Nov 1998, 90

⁵ Barb Cole-Gomoski, "Techies want more than pay", *ComputerWorld* 29 June 1998, n.p., on-line, Internet 2 Jan 1999, available from <http://www.computerworld.com>

⁶ "About Apache", *Apache Web Server Site*, n.p., on-line, Internet 14 March 1999, available from http://www.apache.org/ABOUT_APACHE.html

⁷ Michael Tiemann, *Open Sources*, (Sebastopol CA: O'Reilly Associates), 67

⁸ François Bar, Michael Borrus, Richard Steinberg, "Islands in the Bit-Stream: Charting the Interoperability Debate", *BRIE Working Paper #79*, 1995, n.p., on-line, Internet 2 January 1999, available from <http://www-leland.stanford.edu/~fbar>

⁹ Sharon Gillet, Mitchell Kapor, "The Self-governing Internet: Coordination by Design, Coordination of the Internet", *MIT Computer Science Department Web Site*, n.p., on-line Internet 19 March 1999, available from <http://ccs.mit.edu/CCSWP197.html>

¹⁰ C. Shirky, "In Praise of Evolvable Systems", *Shirkey Systems Web Site*, n.p., on-line Internet 20 Dec 1998, available from <http://www.shirky.com/Articles/>

¹¹ Bruce Caldwell, "Study: Inefficiencies Account for IT Spending Disparities", *Techweb Technology News*, 24 September 1997, n.p. on-line, Internet 11 November 1998 available from <http://www.techweb.com>,

¹² "The Age of Fatherless Code", *Cygnus Corporate Web Site*, n.p., on-line, Internet 11 November 1998, available from <http://www.cygnus.com/news/eracode.html>

¹³ Tom Davey, "IT's Hiring Challenge", *Information Week*, n.p., on-line, Internet January 19, 1998, available from <http://www.infoworld.com>

¹⁴ Patricia Obernorf, "COTS and Open Systems", via email, Software Engineering Institute, cots@sei.cmu.edu, 1998

¹⁵ "HTML Home Page", *World-Wide Web Consortium Web Site*, n.p. on-line, Internet 20 April 1999, available from <http://www.w3.org/MarkUp/#previous>

¹⁶ Margaret Quan, "Networks drive 7% rise in DOD forecast", *EETimes*, 26 October 1998, n.p., on-line, Internet 7 December 1998, available from <http://www.cmpnet.com>

Chapter 4

Applying Open Licensing to Government Software Acquisition and Maintenance

Anything purchased by the military must go through the Department of Defense acquisition system and are subject to the legal agreements imposed by that process. Acquisition is where the rubber meets the road, licensing-wise, and the adoption of open licensing must be integrated into the process. Any change to the current acquisition process should offer the promise of shorter schedules, improved productivity, increased quality or all of the above. The acquisition process recognizes several distinct categories of software systems, each of which will be examined in detail to determine and where open licensing might be appropriate. A view of the applicability of open licensing using the DoD Acquisition Phases as a framework is included as Appendix B.

Commercial Off the Shelf (COTS)

The Department of Defense has suffered some 90% of the federal position layoffs and has had little choice but to improve its efficiency. As the military has downsized and sought to reduce costs it has increasingly adopted the use of COTS- that is packaged software which might be purchased from a retailer. As part of the National Performance Review, President Clinton in 1994 directed the executive agencies to "Increase the use of commercially available items where practicable.." ¹ COTS offers the promise of reduced acquisition cost, shortened acquisition cycles

and increased military leverage on commercial investments.² Open licensing might help hold down costs and obviate some of the obstacles which have been raised to the long-term use of COTS.³

As Isaac Newton observed, for every action, there is a reaction. There are some aspects of COTS which have to be accepted to take advantage of its efficiencies. First, the Government must accept the terms of typical software licenses. With most commercial software you get a license to operate the compiled application. There's usually no information on the configuration, data formats, communication protocols, limitations or bugs present in the system. Unlike custom designed software, there's also no guarantee of the suitability or functionality of the application. Pending legislation, in the form of Article 2B, may impose further restrictions such as exemption from consumer protection laws, legal remedies and notification of restrictions prior to sale.⁴ Since traditional COTS does not offer access to source code, the government is subject to the myriad of licensing restrictions and the maintenance schedule of the vendor over which the government has little control. In addition, the government is often asked to pay for functionality which is of little use or conversely, to accept products which do not entirely meet its needs. As the government becomes an increasingly small percentage of commercial vendors' business, it must examine whether it is possible to have an effective voice in the functionality of the COTS.

Data access will become a major issue as the government examines the long term viability of COTS. Traditionally, software applications have consisted of a large amount of logic for producing and manipulating a relatively small amount of data. Many modern systems, such as web servers, use very small amounts of code to distribute or manipulate truly huge amounts of

data.⁵ Data access has always been a problem and will become one of critical importance as medical records and other information which require long-term access become entombed in obsolete or non-functional, undocumented, proprietary formats. The government cannot afford the continual maintenance effort required to convert data to the latest format every few years.⁶ Without open licensing, the question of how the government will correct problems in products abandoned by the commercial market will become a critical one.

Examining this objection under the auspices of open licensing however, changes the situation substantially. Under open licensing, the government is no longer tied to the vendor if immediate, mission essential fixes are required. Moreover, there are several resolution strategies available including: in-house development, outsourcing the original vendor or using an aftermarket support vendor. Manuals can be easily augmented with the changes or organization and mission specific information. Data formats can be designed or modified to accommodate the government's existing infrastructure or emerging requirements.

The obvious impediment to the proposal to encourage open source licensed COTS is the traditional software industry's tight hold on the source code of their systems. How reasonable is it, one could ask, to expect to obtain functional open source products or convince existing vendors to adopt these provisions? There is evidence that the answer to this question lies in the evolving nature of the software dependent industry and revolves around the question of whether software is a product or service industry.

Industry is moving away from emphasizing the software "product". The new service-oriented members of the information technology industry emphasize the effect of incorporating

the software within an enterprise. Economics is driving this decision. The software corporation Intuit, for instance, now offers a traditional COTS income tax product for \$49 or a web-based, but functionally equivalent, alternative service for \$19.⁷ Within the commercial sector, the software industry is beginning to be dominated by service oriented corporations.⁸ IBM and Digital Equipment Corporation (DEC), for instance, receive a majority of their revenues from support. This service orientation is especially true among start-ups which have sprung up since the widespread availability of the Internet. DejaNews, America On-line and Yahoo! are information technology corporations which market a service. It's interesting to note the extent to which these corporations both depend on and develop open source software. Yahoo! for instance uses open source extensively and attribute to it much of their success.⁹

The military views COTS as a means to maintain parity of functionality, improve interoperability and reduce fixed expenditures. This perspective however, assumes software is a product not an information generation, distribution, manipulation and archival service. A software product may be purchased and once used remains static until the next purchase. Software as a service becomes an living organism evolving along with the organization to meet emerging needs and to eliminate obsolete functionality and data.

Open licensed systems provide increased flexibility and improved supportability. If the source of a manual is available, that content can be tailored to suit individual departments. Open licensed applications can run on every hardware and operating system used in a business; open licensing allows a single code base to be used across the enterprise. Just as many information technology managers believe using a single operating system or application across an

organization will allow the manager to cut costs, many developers appreciate having to maintain and support only one code base. Many organizations are finding shrink-wrapped, open licensed systems can provide both the monetary and schedule benefits of COTS and the flexibility of customized applications.¹⁰ Commercial vendors, such as CISCO Systems, are already utilizing open licensing because vendors, such as Cygnus, are willing and able to conduct prompt, customized development specifically for their needs.

Customized Packages

Customized packages bridge the gap between "pure" COTS and custom developed systems. Customized systems are a much larger category than might first be expected. Much of what is traditionally referred to as COTS- email systems, databases and accounting systems for instance, might more accurately be regarded as customized packages. For example, a word processor may require a set of templates specific to an industry or client. A customized package may also be composed of several COTS packages combined to provide new functionality. For example, commercial credit card processing software might be combined with a commercial web server to provide an on-line store. Most multi-user COTS systems require substantial work before they are functional; the division between COTS and customized packages should be portrayed as a soft grey line. The difficulty is that these adaptations are often undertaken in an ad hoc fashion with little documentation of either the means by which the package can be modified nor the final configuration of the system. The source and extent of the changes are often not available or the changes are available but not documented or understood. Open licensing, if enacted on

subcontracts for in-house maintenance and customization will ensure that the source of changes is at least available. Understanding and maintaining the code base requires effort, but the peer review available under open licensed development should help ensure an adequate base of maintenance developers are familiar with the package. However, if a package requires extensive modification to suit an enterprise or a proprietary advantage is desired, then the organization may elect to write something from scratch. This category of software is called customized automated information systems.

Customized Automated Information Systems

Customized Automated Information Systems (CAIS) are those software system implemented for a specific business environment or mission. Many industries such as banking and military organizations regard their processes and practices as too specialized or proprietary to entrust them to COTS. For these or other reasons, customized automated information systems are produced to meet an organization's special needs and desires. Because CAIS are not normally intended for resale, they offer a fertile ground for open source licensing. However, due to their utility and the amount of effort invested in their development, organizations may be reluctant to release them.

The government is not burdened by profit considerations, it should be free to release CAIS provided that the terms under which they are developed permit this. The rights to software developed by government employees is, of course, held by the government so open licensing is an easy exercise. Systems and documentation developed by outside contractors may sometimes be subject to licensing restrictions. Normally, such licensing considerations don't even enter into

the discussions of in-house software development. In all likelihood, the question of who owns the redistribution rights to a system may not be carefully considered since neither the government nor the development contractor normally envisions reselling or redistributing the system. In this case, discussing open licensing should clarify who owns the rights to the applications. The government may be surprised to find out many consultants hold copyright to code it thought was public property.

This review of open licensing issues may also motivate an organization to modify the management of its development resources. Most organizations feel that people are the key factor in determining whether or not a technology project is considered a success. A review of how those people are organized may be time well spent.¹¹ For instance, such a review may reveal an organization or manager which is hoarding code or documentation where others can't take advantage of it. The code base is the key transport medium for spreading experience and interoperability throughout an organization; it may be seen as the compound interest of information technology. Through the release of source code and documentation, an organization may find that what they thought was a unique requirement is actually shared with many others. Increasing the size of those interested in a CAIS system not only can amortize development costs but allow the bazaar to speed development and improve quality. It may even result in a marketing opportunity for those involved. Sun Microsystems found, for instance, that the release of Java source code resulted in many bug fixes and new operating systems being supported. These contributions increased Sun's goodwill and the value of their Java product line.

Military Unique Development

Military unique development accounts for an increasingly small share of the overall software used within the military but is, perhaps, the most important category to national security. Systems within this category include many major weapons systems with embedded software such as jam resistant communications gear and systems for military unique functions such as nuclear blast simulations.

Many senior leaders assume military unique software will be far buggier and less feature-full than many of its commercial counterparts. For practitioners of the bazaar, this is an expected result; military software is often available to only a small group of persons and not seen or reviewed by outside experts. There's a lot of evidence to suggest that one of the best means of achieving higher quality is the open license distribution of these military systems. As Eric Raymond's paper, "The Cathedral and the Bazaar" so eloquently discussed, ".. given enough eyeballs, all bugs are shallow.." The more talent which examines a system, the greater its rate of progress and level of quality. Unless a strong technical case can be made that unlimited distribution of source code will harm national security it is in the government's interest to ensure all its systems are released under an open licensing arrangement.

This discussion has shown open licensing is appropriate to all of the software systems types in popular use within the government. Moreover, the adoption of open licensing brings with it substantial benefits to the government, in the form of reduced costs and risks, more tailored and efficient solutions and the potential for more reliable and interoperable systems. These benefits stem from unrestricted redistribution and modifications rights and the development bazaar

generally used by open licensed systems. If more military systems were available under open license it would enable one of the most powerful mechanisms for advancing technology transfer and interoperability. The government would lower the risk of archived data being lost. Since source code and documentation are freely available, the lifetime of open licensed systems can be extended indefinitely. Vendors can diagnose difficulties even if they did not originally develop the system. Research institutions can contribute useful ideas and expand on existing functionality to provide something totally new. Commercial firms are more than willing to recognize and take up the maintenance or development of anything for which they see a market. Many past government projects have made the transition to open licensing with excellent results. A few of these instances will be examined in Chapter 5, *Success Stories*.

¹ "Title 3 - Executive Order 12931 of October 13, 1994", Section 1, Paragraph (d), *Federal Register*, Vol 59, No. 199, 17 October 1994

² Paul Kaminski, "Prepared Statements to the Research and Development Subcommittee, House National Security Committee 28 March 1995", *Defense*, Issue Volume 10, Number 46, n.p., on-line, Internet 23 November 1998, available from <http://www.dtic.dla.mil/defenseink/pubs/di/>

³ Dr. M.R. Vigder, J.C. Dean, *Managing Long-Lived COTS Based Systems*, Research Council of Canada, as presented to the Software Engineering Standards Workshop, Monterey, California, 10-13 August, 1998

⁴ White, J.J. & Summers, R.S., *Uniform Commercial Code (4th Edition) Practitioner Treatise Series*, (Eagan MN: West Publishing Corp, 1995), 44

⁵ Tim O'Reilly, Ester Dyson, "Open Minds, Open Source, Version 1.0"

⁶ Simson Garfinkel, "Let My Data Go!", *Hot Wired*, n.p., on-line, Internet 2 January 1999, available from <http://www.hotwired.com>

⁷ "Turbo Tax", *Intuit Corporate Web Site*, n.p., On-line, Internet 2 January 1999, available from <http://www.intuit.com/>

⁸ Mark Gimeinm, "The Next Generation", *The Industry Standard*, 31 December 1998, n.p., on-line, Internet 2 January 1999, available from <http://www.idg.com>

⁹ Tim O'Reilly and Ether Dyson, "Open Mind, Open Source, Version 1.0"

¹⁰ James Moloney, "Success Stories", *Cygnus Corporate Web Site*, n.p., on-line, Internet 30 October 1998, available from <http://www.cygnus.com/success/cisco.html>

¹¹ Kweku Ewusi-Mensah, "Critical Issues in Abandoned Information Systems Development Projects", *Communications of the ACM*, September 1997, 74

Chapter 5

Success Stories

If you claim to be an expert on modern information systems and you aren't rich then most people will infer that you are stupid.

—Philip Greenspun

Open licensing is not a new nor untried idea. The bazaar has contributed many "concepts" such as the world wide web to society. Open license success stories have made possible what one of the founders of CISCO termed, "..the best investment the government has made since the Louisiana Purchase".¹ Open licensed applications are the foundation of the Internet, they dominate important market segments which includes web and mail servers. At the heart of all these innovations are simple but brilliant ideas simply and reliably implemented. Once implemented, the ideas are gently improved, scaled and reutilized to accomplish new goals. The validity of this approach is shown by the long-term durability of their creations. The most popular and longest lasting protocols, formats and applications are open licensed.

The most famous or ubiquitous open source creation, TCP/IP is nearly two decades old. TCP/IP is the communications scheme used by computers on the Internet. The Department of Defense recognized the importance of communications when it funded the creation of TCP/IP, a protocol which allows computers to share information across a network, and its physical manifestation, the ARPANET, to facilitate the spread of ideas and applications among academic

institutions and departments. Visionaries within the Government standardized on this protocol in 1983, well before any widespread commercial recognition.² In contrast to largely unspecified protocols such as Microsoft's original network protocol, Net Beui, TCP/IP has always been fully specified and its implementations were always freely distributed. Problems due to small areas of ambiguity in the specification are avoided since the University of California at Berkeley released some of the earliest versions with its operating systems in the 1980s. Vendor's have incorporated that code into their product avoiding interoperability problems in a way impossible with closed source software or protocols. As a result of years of unrestricted distribution, testing and review, thousands of developers have allowed TCP/IP to become efficient and reliable.

Another product which dominates its market segment also came from Berkeley, the email server known as Sendmail. Sendmail is responsible for carrying nearly 90% of email. It's not only an open source application but the core product for a successful corporation. Sendmail, like TCP/IP is a derivative of work done to take advantage of early connections to the ARPANet.³ Sendmail's adopted protocol, the Simple Mail Transfer Protocol or SMTP, grew quickly to become the lingua de franca of interoperable mail. In 1997 Sendmail was taken commercial by its author, Eric Allman who still allows the application to be freely distributed in source code form. His corporation was funded in part by corporations selling email marketing lists and service. This is an excellent example of how open licensing and the use of software as a service intersect. From research curiosity to popular application to a commercial enterprise and an entirely new market segment, the evolution of Sendmail demonstrates the range of benefits available from open licensing.

One other application which has received a large amount of press is the most popular web server. This web server is the first server to hold over 50% of the world-wide market share.⁴ It's the Apache web server, widely recognized as the most featureful, reliable and innovative of the web server offerings which includes Microsoft's Internet Server, Netscape's Commerce Server and Oracle's Web Server. In fact, the Apache web server is so dominant, Apple and IBM have both decided to provide commercial support for it and make it the default web server on some of their commercial platforms.

How Open Licensing Breeds Success

Open licensed systems dominate the networking and infrastructure segments of the software industry. The use of open-licensed source code has become ubiquitous within even commercial systems. The Request for Comment System used to develop and document the requirements for internet systems has become the model, developing interoperability among all information systems. Developers of open licensed systems were the first to recognize and exploit computers as an information sharing tool.

This chapter has examined only three of the open source systems which without advertising budgets, commercial support or huge paid development staffs have come to dominate their markets. There are many others such as the Domain Name Service which allows us to easily address the millions of computers on the internet. In fact, every aspect of the internet relies on open licensing. These systems succeeded, not because they hid their implementation behind closed doors, but resulted when the developers opened them wide and invited the world to use,

examine and critique their work. This open, widespread analysis led to greatly improved functionality, quality and security.

What was the fundamental reason why these systems, which anyone can download for free generate such a revolutionary change in our life-style and how can something which is freely distributed become the basis for an economic engine? As John McAfee the founder of Network Associates puts it, "If you have two equal products, the one which is free will dominate".⁵ Just as the open license application known as the NCSA HTTP server and its companion browser "Mosaic" led to the popularization of the world-wide web and the internet, these other open source systems have served as the foundation for billions of dollars of economic growth and American prosperity. Information technology corporations are recognizing that the future growth path is in services.

Services are perishable, ephemeral, helpful acts or accommodations.⁶ The knowledge which is used to convey a service is often termed "Intellectual Property". Intellectual Property that, as Scott McNealy of Sun Microsystems puts it, "has the shelf life of a banana".⁷ Nobody wants yesterday's newspaper or last year's software. About a quarter of software systems are killed during development.⁸ If the products of that labor are not put to use, that effort was wasted. Not only were no sales generated but the talent used to produce the code was prevented from working on something else which might have succeeded. Finally, a failed project seldom increases the morale of management or the technical staff. Unless corporations can find a way to

share and evolve last year's software and the by-products of failed systems they'll be forced to spend part of each year rewriting new versions of it into their new offerings.

Open licensing can help organizations to avail themselves of previous efforts. Open licensing removes the barriers to widespread adoption of new or improved ideas and enhances the value of existing systems by allowing organizations to incorporate segments past efforts into new offerings. Open licensing allows you take what you or anyone else has done and tailor it in whole or part to fit a new situation or fix a new problem. Traditional software licenses force the reimplementations of many equivalent systems from scratch. Open licensing allowed groups of cooperating developers to compete with and win against products from the world's largest and most respected firms. The wider adoption of open licensing may be held back from greater contributions mainly by misconceptions.

Chapter 6

The Last Half Mile

Humanity needs practical men, who get the most out of their work, and, without forgetting the general good, safeguard their own interests.

—Eve Curie

There's a term in the telecommunications industry called, "the last half-mile" which refers to all the details required to get even simple things to work. A communications network stretching halfway around the world can be stopped in its tracks by a neighborhood council preventing the company from digging a trench across a road leading to the building you want to connect. If the Department of Defense wishes to implement open licensing, what "neighborhood councils" exist which might stymie the effort?

The framework of open licensing for government documents is already in place for unclassified information. The Freedom of Information Act lays out the circumstances under which the Department of Defense must divulge information and redistribution of the released documentation is not generally restricted. Information systems and the data they generate and store are partially covered by the Freedom of Information Act, however the source code of those applications must be addressed separately. There are no major legal or policy obstacles to a

decision to formally decide that software and data wholly funded by the government should be open licensed. Open licensed products are currently used by many operational units and there are no major obstacles expanding this use. The major question in this area is, of course, whether or not such terms would be acceptable to commercial organizations.

Improved quality and reduced development costs are the major incentives offered by open licensing. Misunderstanding the nature of software source code and the value of a review of that code by others are the major impediments to more widespread commercial adoption. While there are no documented examples of corporations whose business has been damaged by the release of source code, corporations will naturally be hesitant to modify their current policies. Software system source code is often regarded as different and more powerful than other forms of documentation. Despite the real-world success of open licensing, it is still difficult to convince government or corporate management that there is no real difference between source code and other forms of documentation. Many senior executives have not had the opportunity to create software and are often uncomfortable with the developers who do. Probably only financial success and market share will convince many executives and managers to consider open licensing. Rather than propose any decree or coercive program, the growing market emphasis on open licensing should provide sufficient incentive to migrate corporations in this direction.

As other forms of science, peer reviews and testing of computer software are the most effective tools for ensuring robust and highly functional applications.⁹ During major upgrades and development of software intensive systems, such reviews can significantly lower risk.¹⁰ As a specific example of the benefits of open licensing and development within the bazaar, consider

how difficult it would become for developers to hide design and implementation faults. Reviews within a bazaar should not be confused with the more traditional "Independent Verification and Validation" efforts of the past. They are instead an extension of the process utilized by journals of hard science. Independent verification and validation efforts are not conducted by a nearly unrestricted number of people with a vested interest in the success of the system. Under an open licensing arrangement, code may be reviewed by hundreds or even thousands of independent eyes each of which is focused on those areas of the system where they are a functional expert or interested party. The Apache web server for instance, is developed by a core team of about 20 individuals. Contributions, enhancements and bug fixes however, have been submitted by thousands of users and developers around the globe.¹¹

The question of whether there should be a limited, selected audience of peers or a thoroughly open process such as used on the Internet is a controversial one. Experiences with many thousand of applications tend to indicate that the more open this process is, the greater the benefits. In general, those individuals taking the time to review and perhaps augment the information will be knowledgeable and mature enough to serve as an supplement rather than a detriment to the process. There are particular advantages to the release of such artifacts as documentation on a continual rather than periodic basis. Allowing continual comments rather than massive periodic reviews increased the likelihood that controversies will be caught and fixed early when such changes are relatively inexpensive. The increased use of electronic documentation can be utilized, allowing field comments to be shared and discussed among the end-users of the information. This will require acquisition and headquarters commands to release

some control over these documents, trusting them in the hands of Joint Forces and tactical echelons of command.

There are several areas where the wider adoption of open licensing would offer clear benefits. Research organizations, such as government laboratories, should be actively encouraged to adopt open licensing and utilize open license development methods. Systems released in this fashion can easily be adopted by commercial industry for commercial development. Another low hanging fruit software category is security enforcement and detection. The National Security Agency's FORTEZZA algorithms used by the Defense Messaging System and the Defense Advanced Research Project Agency's firewall toolkit are two extremely successful examples of how open licensing can dramatically increase the acceptance and quality of systems in this area. The growth of vendors such as Cygnus and the many distributors of the Linux and FreeBSD operating systems offering shrink-wrapped, open license products show this business model offers significant opportunities. The widespread adoption of open licensed systems by commercial institutions, large and small, indicates they feel such systems may reduce costs or provide greater productivity than other commercial offerings.

The Department of Defense should move to actively adopt open licensed systems as much as possible. The common operating environment for instance, should expand its use of open licensed software to include open source operating systems such as Linux and FreeBSD. Servers such as the Apache web server and Sendmail email router should be officially adopted. Greater encouragement should be given to existing, successful open licensed government efforts such as the GRASS system which is widely used for hydrological and topological modeling.¹² While the

department has fostered many of the initial open licensed work the success of those systems makes it clear that there are many benefits to be gained from its expanded use. Cost advantages, interoperability and quality all stand to benefit from a greater use of open licensed software, particularly as more and more open licensed systems are available in "shrink-wrap" ready to use form. Where a ready made package is not available, the Department of Defense should consider the use of a bazaar to develop military specific systems. This method, which expands the modern notions of incremental development methods to their logical conclusion, acknowledges the ongoing nature of software intensive systems. This promotion of a bazaar for obtaining our future software needs will not only bring benefits to Department of Defense programs but hopefully increase the retention of our valuable information workers by providing challenging work they can use to demonstrate their work and improve their reputation. If the Department does not foster such a bazaar of experts able to maintain and support its systems, COTS and specially developed, it runs the risk of owning more and more unsupportable systems or paying exorbitant costs to sole-source suppliers.

¹ Robert X. Cringely, *Nerds 2.0.1*, National Public Television, 1998

² Craig Hunt, *TCP/IP Network Administration*. (Cambridge, MA: O'Reilly & Associates, 1992), 2

³ Tim O'Reilly and Ether Dyson, "Open Mind, Open Source, Version 1.0"

⁴ "Netcraft Web Server Survey", *Netcraft Survey Web Site*, n.p., on-line, Internet 7 February 1999, <http://www.netcraft.com>

⁵ Robert X. Cringley, *Nerds 2.0.1*

⁶ *Websters Encylopedic Unabridged Dictionary of the English Language*, (Gramercy Books, New Jersey, 1996)

⁷ Scott McNealy, from his speech presented at the "Converging on Opportunity Seminar", San Francisco, June 1998.

⁸ Tom Demarco, *Peopleware*, (New York, NY:Dorset House Publishing Corporation, 1992), 4

⁹ John Dominque and Paul Mulholland, "Fostering Debugging Communities on the Web", *Communications of the ACM*, April 1997, 65-71

¹⁰ Kweku Ewusi-Mensah, "Critical Issues in Adandoned Information Systems Development Projects", 76

¹¹ "Apache HTTP Server- Regular Contributors", *Apache Web Server Site*, n.p. on-line, Internet 14 Marcy 1999, available from <http://www.apache.org/contributors.html>

¹² "Grass GIS Homepage", *GRASS Web Server Site*, n.p. on-line, Internat 20 April 1999, available from <http://www.baylor.edu/~grass>

Appendix A

The Open Source Definition (*Version 1.0*)

Open source doesn't just mean access to the source code. The distribution terms of an open-source program must comply with the following criteria:

1. Free Redistribution

The license may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of downloading the source code, without charge, via the Internet. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code.

The license may restrict source code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups.

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor.

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License.

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product.

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Contaminate Other Software.

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. Example Licenses.

The GNU GPL, BSD, X Consortium, and Artistic licenses are examples of licenses that we consider conformant to the Open Source Definition. So is the MPL.

Bruce Perens wrote the first draft of this document as 'The Debian Free Software Guidelines', and refined it using the comments of the Debian developers in a month-long e-mail conference in June, 1997. He removed the Debian-specific references from the document to create the 'Open Source Definition'.

Eric S. Raymond

Appendix B

Adoption of Open Licensing

This appendix will discuss how the acquisition system or an operational unit might utilize open licensing. Since the traditional model of acquisition is familiar to most government officials it will be used as the framework for this discussion. The acquisition phases from DoDD 5000.1, Para D.3.e and DoD 5000.2-R Part 1.2 are used as a framework. Both of these documents are available in the web edition of the Acquisition Deskbook on *<http://www.acq.osd.mil>*.

Phase 0: Concept Exploration

During Phase 0, the beginning of a program, the enthusiasm of the commercial and government members of the team is often very high. The integrated product teams are looking for ways to increase innovation, efficiency and reduce costs at the point where such decisions can have the most impact on total program costs. The relatively few contract vehicles used during this stage are flexible enough to accommodate open licensing and there are fewer barriers to the adoption of existing open licensed systems in partial or total fulfillment of the sponsors' needs. The use of open licensing and bazaar type development processes offer an opportunity to significantly reduce cost and schedule. As prototypes and ideas are released, their open source

nature makes it more likely that commercial or government organization might adopt or build upon your initial product. Conversely, if a program actively searches for open source to use, it's likely at this stage that suitable application or components can be incorporated into the development. Obviously, the more pervasive this adoption the less development, cost and schedule will be needed to fulfill the sponsor's total requirement.

How can the use of open source systems be encouraged? During Phase 0, the government often releases a series of broad area announcements or other pre-solicitation documents. Within these documents mention should be made of the government's desire to utilize open licenses. This will ensure vendors are considering this option and allow them the opportunity to submit questions on the matter if they are unfamiliar with the concept.

Of course, any software development conducted by the government or supporting contractors should be released under open licensing and other vendors and organizations encouraged to review, comment on or enhance the source or documentation. Ideally, this process will result in a positive circle of support. This process is very similar to that used by Research and Development organizations such as the Defense Advanced Research Project Agency and the success in that realm has been reviewed in the main body of this document. Unfortunately, not all programs make it past Concept Exploration and sometimes important work is buried along with the program. A decision to terminate or postpone a system at the conclusion of Phase 0 will, if done under the auspices of open licensing, still allow the release in components which might be utilized without restriction by future programs or enhanced in the commercial sector. This feature of open licensing may help relieve some of the typical concerns over Phase 0

expenditures which, if not further developed, might have resulted in proprietary or undocumented designs not suitable for follow-on use.

Phase 1: Program Definition and Risk Reduction

Phase 1 may follow Phase 0 Concept Exploration or be itself the beginning of a system based on existing ideas or products. By its nature, Phase I involves testing the ideas or existing systems and produces a further defined or enhanced design for use by Phase 2. Often, several competing versions of a system are refined during this phase. This parallel development is used as risk-reduction strategy. Open licensing also encourages the initial exploration of many avenues to fulfill a requirement and encourages the mixing and matching of ideas to reach an optimal solution. It's clear that the Phase 1 parallel development strategy precisely models the underlying philosophy of open licensing and the bazaar development model most commonly utilized by open licensing proponents. Open licensing will also facilitate the cross-fertilization of ideas among parallel products by removing licensing barriers and encouraging a thorough examination of the preliminary designs, documentation, data formats and implementations.

Several small award - fee or fixed-price development contracts are typically utilized during Phase I. If open licensing is adopted as part of this strategy, clauses facilitating the exchange of information among participating integrated product teams should be included as well as incentives for those parties using ideas and products of other teams or contributing substantial innovations or contributions themselves. Such incentives and agreements are often included in current Phase 1 contract vehicles since the use of open licensing adds synergy to the idea and is an appropriate vehicle for utilizing the benefits originally envisioned.

Phase 2: Engineering and Manufacturing Development

Following Phase 1, there is often a "downselect" of the parallel efforts and one hybrid approach is chosen for further refinement. Phase 2 is characterized by the often difficult work of turning a prototype into something more suited for production or deployment. Though some estimates show 70% of the total systems costs are determined during Phase 0, Phase 2 attempts to reduce costs and improve performance within the bounds of the chosen design. Documentation, detailed specifications and testing procedures for verifying system performance are also produced during this phase.

This phase may become a lengthy one and difficulties may arise as more and more of the program content becomes proprietary or known only to a select group. If not properly managed this phenomena may result in a sole source situation where one vendor holds the government captive for the remainder of the system's lifetime.

If a system is not released under the terms of open licensing, it is difficult for a single vendor or consortium to adequately design, implement and test a system. Conversely, a system released under open license at this stage can be easily shared or augmented through agreements with other vendors and of course shared with our Allies. Other agencies within the government or commercial organizations may begin to develop the system in parallel for other purposes or commercial resale. Since this is the first phase where the product might be useful as a weapon system, the release terms of certain systems or components of systems might need, in the interests of national security, to be subject to distribution limitations.

The mention of distribution restrictions brings the issue of security into the discussion. A common question concerns the effect of open licensing on the security of a system. Security breaches can result from a failure to follow instructions or procedures or from unexpected situations or unexplored exploitations of an implementation or design. While many non-technical managers believe the release of source code lowers the security of a system, experience shows the opposite. Security "holes" are omissions or weaknesses designed into the code. Compiling the code into a binary application doesn't fix a security hole or hide it from prying eyes.¹ Unfortunately, the discovery of all security problems is known in computer sciences as an NP-Hard problem- that is one which is believed to be impossible to resolve absolutely. The best current practice can hope for is to avoid obvious mistakes, test as much as a project's budget can afford and correct problems as soon as they are identified. Open licensing is designed to tackle this process. Most security experts believe that the release of source code improves, rather than diminishes the security of a software system. For instance, following the National Security Agency's release of the Skipjack encryption algorithm, researchers discovered serious flaws in the system and proposed a solution to the problem.² The error might easily have been exploited without our knowledge but it is unlikely that anyone doing so would have proposed a fix and it would be difficult for the National Security Agency to admit such a flaw. Open licensing of the algorithm provided an avenue for both the unbiased review of the algorithm by experts and a quick and effective resource. As the saying goes, bad news doesn't improve with time. It is far better to go "open kimono" and identify security risks early when there is a better chance they can be fixed with less schedule and cost risk. Further, if components supplied by foreign or

unfamiliar subcontractors are incorporated into the system, open licensing makes it far less likely that an accidental or deliberate security problem will be introduced.

When a fix is identified, the government must also work to ensure the fix is incorporated as soon as possible not only into the system within which the problem was identified but anywhere else the implementation may have been used. A bad encryption algorithm for instance must be replaced in all systems which use it be they phones, email systems or radios. Open licensing doesn't alleviate the need to test or the difficulty of this problem but it raises the possibility that someone, somewhere will discover and report the problem before the system goes operational and the consequences of the security problem could be life threatening. Open licensing also ensures that an identified fix can be incorporated without being hindered by licensing arrangements or proprietary agreements. If one of configuration or operation and procedures are required to prevent a vulnerability, open licensing allows the manuals and technical orders to document the vulnerability.

It is understandable that the government might want to restrict distribution of source code. For instance, the government may wish to limit the distribution of important command and control systems to those Allies and Defense Partners to whom we wish to distribute this combat advantage. If the government were able to produce, for example the "perfect" firewall it is unlikely the government would wish its adversaries to possess such a valuable tool. So, the circle of redistribution for government systems may need to be restricted for national security reasons. The government though should carefully consider any decisions to restrict redistribution. Legally, it may be that the Freedom of Information Act may be considered to cover software

systems. For those authorized to receive the system though, we should not want to deny their ability to enhance, integrate or debug the system anymore than we would appreciate operational restrictions being placed upon a combatant commander. The analogy of software and information is strong here. Consider a Commander who is informed of the availability of a new piece of intelligence. We would not want to restrict the application of that information if its use would enhance the combat power of the commander's forces. Further, we would expect the commander to synthesize the new piece of intelligence and his existing knowledge to produce a solution more powerful than either individual piece of information would allow.

There are already several successful examples of government sponsored open source projects whose redistribution is restricted. Both the Secure Multipurpose Internet Mail Extensions (S/MIME) funded by National Security Agency's (NSA) X Division and the Public Key Initiative which is funded by the National Institute of Science & Technology (NIST) are classic examples of open licensed projects which promise major advances of the state of the art. The redistribution of the encryption code in both projects is restricted by the United States export controls yet the programs still benefit the government, which receives assistance from commercial and private developers interested in the functionality. The information technology sector in turn gains a functional and efficient security protocol which, since is not controlled by any single corporation or organization may be freely incorporated into their products and will ensure interoperability with other vendors who support the protocols.³

The use of open licensing in Phase 0 and Phase 1 may reap unexpected benefits during Phase 2. This is true because pressure within a bazaar fosters designs and algorithms which

function even when partially implemented. This large numbers of knowledgeable eyes reviewing the source results in a greater percentage of the "engineering details" such as sizing, timing, scaling and capacity being recognized and addressed earlier in the development process.

During Phase 2, work is often conducted with a competitively awarded contract, typically of a cost reimbursement or indefinite delivery/indefinite quantity type. If open licensing is desired, this should be made known to the contractor during the source selection process when they would be most likely to accept the provisions. After award, it is unlikely they would agree to the distribution of their products. The contractor might however, agree to utilize existing open licensed systems or information if doing so would reduce their development costs or risks.

Phase 3: Production, Fielding/Deployment and Operational Support

The results of Phase 2 development efforts are even further refined during Phase 3. During Phase 3 hardware items are produced or purchased, tested and assembled into an integrated system. Further documentation for both users and maintenance personnel is developed as other logistical products such as software support facilities are installed. The Phase culminates in the actual fielding and some initial support of the system. For software intensive products, most of the effort in this phase involves detecting and fixing the various issues which crop up during widespread deployment into the "hostile" real world operational environment. This phase may continue indefinitely as continual improvements and fixes are deployed- especially in the case of products which must maintain currency with the evolving commercial offerings.

The capital investment aspect of Phase 3 is obviously applicable to pure hardware efforts however software burned into or embedded in hardware is subject to the same issues as

traditional hardware. The fielding, deployment and initial operational support of software intensive efforts may be more difficult than for traditional programs. These difficulties are due to the widespread perception that software changes are easily accomplished, the rapid evolution of commercial offerings and the current push towards the use of common application servers and the interoperability problems which develop in such an environment.

If open licensing has not been adopted during previous phases of system development, Phase 3 is probably not an opportune time to begin the transition. Typically, there is not a great amount of competition or modification of the system parameters or design so the dramatic effects of community or bazaar development may not be able to produce their usual results. Further, it is unlikely that either the government or commercial team will want to interrupt what is likely a very tight schedule to adopt anything new.

Opportunities to incorporate open licensed components into the testing and certification processes still exist though. A open licensed implementation of nearly all common protocols exists and these are excellent "test stubs" which may be used to ensure the interoperability of the interfaces within the system under development. Further, if documentation is released or portions of it developed under an open license then documentation costs, which may comprise a large percentage of system costs, may be dramatically reduced by incorporating the end users into the review and comment phase. This setup can avoid the usual rounds of expensive comment periods and endless debate. Open licensed documentation may also be used to create versions of documentation which incorporate organization specific conditions. For instance, a technical manual could be updated to show actual pictures of the facility or the names of points of contact

within the organization. This observation leads to a discussion of how else open licensing can benefit operations.

Operational Support

After the deployment and initial support period ends, the program begins the work which all the previous development supported. This operational period may last many years and during that period the system should ideally operate with a minimum of administration and maintenance costs. Since the system is operational, it is being maintained by the always tight Operation and Maintenance budget. This budget is used for everything from airplane parts to food, expenditures for software operation and maintenance are seldom viewed as a mission-critical item when the "beans and bullets" are in short supply. Since development and production funds are not usually available to operational units, once a system is deployed it may be difficult to upgrade or enhance. Thus, these activities are perennially underfunded and undermanned. Because most of the operation is performed by enlisted technicians and junior officers, there's always a problem recruiting and retaining qualified personnel to operate the systems- especially within today's burgeoning economy. Money, personnel and the local environment are the major contextual elements of this phase.

During this phase of a system's lifetime the deployed system is being maintained and administered for thousands of sites across hundreds of organizations. Each of these installations is somehow unique and may require not only specialized documentation but new functionality. Senior leaders and managers within the government are adverse to these local enhancements because they are perceived as maintenance and procurement cost drivers. Administration and

maintenance cost growth, it's felt, is directly proportional to the amount of variation among local system configurations and functionality.

How does open licensing fit into this picture? Let's consider one very compelling quality of most open licensed systems- they cost little to nothing to procure. Once procured they can be deployed to as many users or locations as required without further acquisition expenditures. Because these systems are designed to be distributed over the internet there are many tools available to ease and automate deployment and upgrades. A fully featured multi-user operating system can be had for unlimited users for a one time cost of \$3.00. A similar version with commercial support costs less than \$50.00 per user. Firewalls, web servers, graphical user interfaces, debugging tools, directories, etc. are all available for similar fees. Secondly, if none of these products fully meets the needs of the organization it's very easy to enhance or adapt the system since full source code is available and there are no legal restrictions on adaptations. The only reason to restrict the scope of these changes are ones of configuration control and quality assurance. This work can be done with government employees or contracted out.

Open licensed products are already widely utilized with or without the knowledge of senior management. Senior non-commissioned officers and personnel under field conditions are expected to have their equipment functional at all times and react to the functional demands of the field commanders they are serving. If funds or a commercial product fulfilling the commander's need is not available, operational units will often use an open licensed system instead. Many times, these products are used simply because they are better than the commercial offerings. As a case in point, the world's fastest Microsoft Windows file server is in fact an open

source system and operates some 200% more efficiently under an open source operating system than under Microsoft's NT operating system itself.⁴ Perl, a computer language which is the darling of the internet and used to "glue" together many famous web sites such as DejaNews is also open source. Administrators can use the open source facilities to add functionality to their sites, to automate common tasks, test and debug installations or even to implement new functionality.

During Operations, the restrictions on the original system may be inherited from the development phases or the licenses of the commercial vendors providing components. There should generally be no restrictions on the addition of open source components. However, the Commander will have to decide if he or she is comfortable with the ramifications of open licensing and the level to which their organization should participate. Further, the organization should prepare procedures and an approval process for evaluating and incorporating open source systems into their operations. Open licensing is intended to improve the quality and reliability of operational systems while reducing costs. Adopting open licensing does not mean that typical quality assurance or configuration management procedures should be circumvented. The Commander should also discuss this proposal with their subordinates to see if they are familiar with the concept and whether or not open licensed products are already in use. If the Commander is inclined to adopt open source then he or she may appoint a few "evangelists" who can further develop a proposal for incorporation of open license components in the organization. A Commander should not be surprised if this is a coveted position. Surveys and studies of Information Technology workers have shown that compensation is a smaller component of

satisfaction than managers might otherwise suspect. Ambitious and enthusiastic workers typically place great value on the extent to which their work is interesting and challenging. "Cutting edge" areas, such as open licensing, are currently held in high regard by many information technology professionals. The opportunity to participate in this area can significantly affect the morale of subordinates, especially if that participation might result in substantial recognition within the employee's organization or the larger information technology community.

As this discussion of the acquisition cycle has shown there are multiple opportunities and broad areas where open licensing can both improve the service provided to Commanders and reduce the cost and risk of warfighting and support systems.

¹ Eric Raymond, "Why Software Developers Benefit from Open-Source Software", *Open Source Web Site*, n.p. on-line, Internet 15 March 1999, available from <http://www.opensource.org>

² Two Israeli researchers, see email

³

⁴ "World's Fastest Windows Server Software", *Linux Weekly News*, n.p. on-Line, Internet, 21 January 1999, available from <http://lwn.net/1999/0121/samba.html>

Glossary

Bazaar	An expression taken from the open air markets of the years gone by where individuals would operate independently to construct a larger organization. In the area of open licensing, the bazaar is used to refer to groups of cooperating but loosely structured user-developers building and using applications and information.
Code Base	The set of files and information used to build a software system or application. A single code base means that the same set of files can generate the same application which can be used on all hardware platforms. Without a single code base developers and users may need to learn more than one application to accomplish the same task.
Debug	The process by which the quality and functionality of software is improved. It is usually regarded as the process by which "faults" are removed from source code.
Developers	Used to refer to the people involved in the design and implementation of software systems. This is not just "programmers" but software architects, user interface artists, technical writers, project managers and the functional experts who help determine the functionality of the final system.
Firewall	A software system used to protect or monitor a subset of a larger network.
FreeBSD	A popular open licensed operating system used by many large internet sites including Yahoo! and DejaNews.
Linux	Pronounced "Lyn-nux". An open source operating system with a large internet following. It is the most popular operating system for hosting internet servers and favored by internet service providers for reliability and features. More information is available at http://www.linux.org
Open Content	A scheme under which the content of documents may be redistributed and altered within a set of legal guidelines. See <i>Open Licensing</i> or http://www.opencontent.org
Open Licensing	A term used to refer to both open source and open content licenses. Open Source licensing addresses software source code specifically

	while Open Content Licensing focuses on the redistribution restrictions on written information. More information is available at http://www.opensource.org
Open Source	A scheme under which the source code of computer systems may be redistributed or altered within a set of legal guidelines. See <i>Open Licensing</i> or http://www.opensource.org
Peer Review	A process by which ones' work is verified by a critical examination by others. Agreement by others in the quality and capability of the work is regarded as an indicator of the validity of the work.
Reproducibility	The ability of others to conduct a documented process and arrive at the predicted results.
Scientific Method	A technique for the discovery and verification of fact. It relies on a tentative description known as a hypothesis which is used to make predictions. These predictions are tested by experimentation and the hypothesis is modified, if required and the verification process repeated until there are no discrepancies between observation, theory and experimental results. It is also widely regarded to depend upon the concept of peer review and reproducibility.
Shrink-wrapped	A term used to describe a documented, ready to use software system suitable for retail sale.
Source Code	The text which is used by software developers to produce an application or software system.
Test Stubs	Small software routines which are inserted or used to quickly exercise software to evaluate its quality and functionality.
User-Developer	The open-source licenses allow users to contribute to the content and quality of the documents and applications they utilize.
Y2K	A term used to describe concerns with software being able to handle the four digit data format required after the year 2000.

Bibliography

- "Company Information". *The Berkeley Software Design Corporate Web Site*, On-line, Internet 20 March 1999. Available from <http://www.bdsi.com>.
- "Intuit Corporate Web Site". *Intuit Corporate Web Site*, On-Line, Internet 2 January 1999. Available from <http://www.intuit.com>.
- "House Report 103-104, 1993 Edition, House Committee on Government Operations". *Library of Congress Web Site*, On-line, Internet 20 November 1998. Available from <http://www.loc.gov/>.
- "Leahy - Ashcroft Amendment to S.1609 to Authorize the Next Generation Internet 26 June 1998". *Library of Congress Web Site*, On-Line, Internet 15 January 1999. Available from <http://www.house.gov/>.
- "Netcraft Web Server Survey". *Netcraft Survey Web Site*, On-Line, Internet 7 February 1999. Available from <http://www.netcraft.com>.
- "The Age of Fatherless Code". *Cygnus Corporate Web Site*, On-line, Internet 2 January 1999. Available from <http://www.cygnus.com/news/eracode.html>.
- "Title 3 - Executive Order 12931 of October 13, 1994 Section 1, Paragraph (d)". *Federal Register*, Vol 59, No. 199, 17 October 1994.
- Bar François, Michael Borrus , Richard Steinberg. "Islands in the Bit-Stream: Charting the Interoperability Debate". *BRIE Working Paper #79*, n.p. On-line. Internet 2 January 1999. Available from <http://www-leland.stanford.edu/~fbar>.
- Burke, James. *The Day the Universe Changed*. Boston, MA: Little Brown & Company, 1986.
- Caldwell, Bruce. "Study: Inefficiencies Account for IT Spending Disparities". *Techweb Technology News*, 24 September 1997, n.p. On-line, Internet 11 November 1998. Available from <http://www.techweb.com>.
- Clemins, Admiral. "IT-21: The Path to Information Superiority, Quadrennial Defense Review Secretary's Report to Congress". *Chips*, July 1997, n.p. On-Line, Internet 11 November 1998. Available from http://www.chips.navy.mil/chips/archives/97_jul/file1.htm.
- Cole-Gomoski, Barb. "Techies want more than pay." *ComputerWorld*, 29 June 1998, On-Line, Internet 2 January 1999. Available from <http://www.computerworld.com>.
- Cringley, Robert X. *Nerds 2.0.1*. National Public Television, 1998.
- Curie, Eve. (translated by Vincent Sheean) *Madame Curie*. New York: Pocket books, Simon and Schuster, 1946.
- Cusuman, Michael and Richard Selby. "How Microsoft Builds Software". *Communications of the ACM*, June 1997, 53-61.

Davey, Tom "IT's Hiring Challenge". *Information Week Online*, n.p. On-line. Internet 19 January 1998. Available from <http://www.infoworld.com>.

Demarco, Tom. *Peopleware*. New York, NY: Dorset House Publishing Corporation, 1992.

Dominque, John and Paul Mulholland. "Fostering Debugging Communities on the Web". *Communications of the ACM*, April 1997, 65-71.

Ewing, Jonathon and Sami Lais . "70 CPUs add up to big power". *Government Computer News*, 7 September 1998, 43.

Ewusi-Mensah, Kweku. "Critical Issues in Abandoned Information Systems Development Projects". *Communications of the ACM*, September 1997, 74-80.

Foster, Edward. "1997 Product of the Year". *Infoworld*, n.p. On-Line. Internet 20 December 1998. Available from <http://www.infoworld.com>.

Garfinkel, Simson. "Let My Data Go!". *Hotwired*, n.p. On-line, Internet 2 January 1999. Available from <http://www.hotwired.com>.

Gillet, Sharon and Mitchell Kapor. "The Self-governing Internet: Coordination by Design, Coordination of the Internet". *The MIT Department of Computer Science Web Site*, n.p. On-line. Internet 19 March 1999. Available from <http://ccs.mit.edu/CCSWP197.html>.

Gimein, Mark. "The Next Generation". *The Industry Standard*, 31 December 1998, n.p. On-line. Internet 2 January 1999. Available from <http://www.idg.com>.

Hughes, Thomas P. *American Genesis*. Middlesex, England: The Penguin Group, Harmondsworth, 1989.

Hunt, Craig. *TCP/IP Network Administration*. Cambridge, MA: O'Reilly & Associates, 1992.

Hunt, Gaillard, ed. *The Writings of James Madison*. Volume III, 1865, New York: G.P Putnam's Sons, 1910.

Hunt, Gaillard, ed. *The Writings of James Madison*, Volume IX, 1819-1836, New York: G.P Putnam's Sons, 1910.

Knuth, Donald E. "Rewards", n.p. On-Line, Internet 20 December 1998. Available from <http://www-cs-staff.stanford.edu/~knuth>.

Knuth, Donald E. "George Forsythe and the Development of Computer Science". *Selected Papers on Computer Science*. Stanford, CA: Center for Study of Language and Information Publications, 1996. 241

Leibovich, Mark. "The Spreading Grass-Roots Threat to Microsoft". *Washington Post*, 3 December 1998, A01.

McNealy, Scott. from his speech presented at the "Converging on Opportunity Seminar", San Francisco, June 1998.

Moloney, James. "Success Stories". *Cygnus Corporate Web Site*, n.p. On-Line, Internet 2 January, 1999. Available from <http://www.cygnus.com/success/cisco.html>.

Obernorf, Patricia. "COTS and Open Systems". 1998. Software Engineering Institute, received via email from cots@sei.cmu.edu.

O'Reilly, Tim and Ether Dyson. "Open Mind, Open Source, Release 1.0". *O'Reilly Corporate Web Site*, n.p. On-Line, Internet 2 January 1999. Available from <http://www.oreilly.com>.

Pendery, David. "News". *Infoworld*, 1 Feb 1999. 16

- Kaminski, Paul G. "Prepared Statements to the Research and Development Subcommittee, House National Security Committee, 28 March 1995". *Defense Issue* Volume 10, Number 46, n.p. On-Line, Internet 23 November, 1998. Available from <http://www.dtic.dla.mil/defense/links/pubs/>.
- Raymond, Eric S. "The Cathedral and the Bazaar". *The Open Source Web Site*, n.p. On-line, Internet 11 November 1998, Available from <http://www.tuxedo.org/~esr>.
- Raymond, Eric S. "Frequently Asked Questions about Open Source". *The Open Source Web Site*, n.p. On-line, Internet 11 November 1998, Available from <http://www.opensource.org>.
- Raymond, Eric S. "Why Software Developers Benefit from Open-Source Software". *The Open Source Web Site*, n.p. On-Line, Internet 15 March 1999, Available from <http://www.opensource.org>.
- Quan, Margaret. "Networks drive 7% rise in DOD forecast". *EETimes*, n.p. On-Line, Internet 26 October 1998. Available from <http://www.cmpnet.com>.
- Shirky, C. "In Praise of Evolvable Systems". *Shirkey Systems Web Site*, n.p. On-line, Internet 11 November 1998, Available from <http://www.shirky.com/Articles>.
- Tiemann, Michael. *Open Sources*. Sebastopol CA: O'Reilly Associates, 1999
- Toole, Dr. Betty. "Ada Byron, Lady Lovelace". *The Ada Lovelace Web Site*, n.p. On-Line. Internet 10 November, 1998. Available from <http://www.scottlan.edu/lriddle/women/love.html>.
- Tzu, Sun. *The Art of War*. London UK:Oxford University Press, 1971
- Valloppillil, Vinod. "Open Source Software, A (New?) Development Methodology". *The Open Source Web Site*, n.p. On-line, Internet 11 November 1998. Available from <http://www.tuxedo.org/~esr/halloween.html>.
- Vigder, Dr. M.R. and J.C. Dean. *Managing Long-Lived COTS Based Systems*, National Research Council of Canada, as presented to the Software Engineering Standards Workshop, Monterey, California, 10-13 August, 1998.
- Websters Encyclopedic Unabridged Dictionary of the English Language*, New Jersey: Gramercy Books, 1996
- White, J.J. and R.S. Summers. *Uniform Commercial Code (4th Edition) Practitioner Treatise Series*, Eagan MN: West Publishing Corp, 1995
- York, Thomas. "1999 IT Career Outlook", *Infoworld*, 30 Nov 1998, 90